DataSet_Analyiss

Paweł Gędłek, Andrzej Szaflarski

13 06 2020

Fifa17 dataset

https://www.kaggle.com/artimous/complete-fifa-2017-player-dataset-global?select=FullData.csv

Content:

- 17,000+ players
- 50+ attributes per player ranging from ball skills aggression etc.
- Player's attributes sourced from EA Sports' FIFA video game series, including the weekly updates
- Players from all around the globe
- URLs to their homepage
- Club logos
- Player images male and female
- National and club team data

Columns:

- 'Name'
- 'Nationality'
- 'National_Position'
- 'National_Kit'
- 'Club'
- 'Club_Position'
- 'Club_Kit'
- 'Club_Joining'
- 'Contract_Expiry'
- 'Rating'
- 'Height'
- 'Weight'
- 'Preffered_Foot'
- 'Birth_Date'
- 'Age'
- 'Preffered_Position'
- 'Work_Rate'
- 'Weak_foot'
- 'Skill_Moves'
- 'Ball_Control'
- 'Dribbling'
- 'Marking'
- 'Sliding_Tackle'
- 'Standing_Tackle'
- 'Aggression'
- 'Reactions'
- 'Attacking_Position'
- 'Interceptions'
- 'Vision'
- 'Composure'
- 'Crossing'
- 'Short_Pass'
- 'Long_Pass'
- 'Acceleration'
- 'Speed'
- 'Stamina'
- 'Strength'
- 'Balance'
- 'Agility'
- 'Jumping'
- 'Heading'
- 'Shot_Power'

- ‘Finishing’
- ‘Long_Shots’
- ‘Curve’
- ‘Freekick_Accuracy’
- ‘Penalties’
- ‘Volleys’
- ‘GK_Positioning’
- ‘GK_Diving’
- ‘GK_Kicking’
- ‘GK_Handling’
- ‘GK_Reflexes’

Początkowo dane należało wyczyścić: m.in.: - dodać kolumny z skonwertowanymi danymi stringowymi na integery za pomocą techniki onehot encoding,

```
Fifa <- read.csv("Fifa17_ext.csv", header = TRUE, na.strings = "?")
Fifa1h <- subset(Fifa, select = -c(X, Name, Nationality, Club, Club_Position, Club_Joining, Birth_Date, P
reffered_Foot, Preffered_Position, Work_Rate))
na.omit(Fifa1h)
dim(Fifa1h)
```

```
attach(Fifa)
head(Fifa1h)
```

Regresja ratingu zawodnika względem wieku

Jako pierwszą postanowiono przeprowadzić analizę wpływu wieku zawodnika na jego rating.

Regresja liniowa

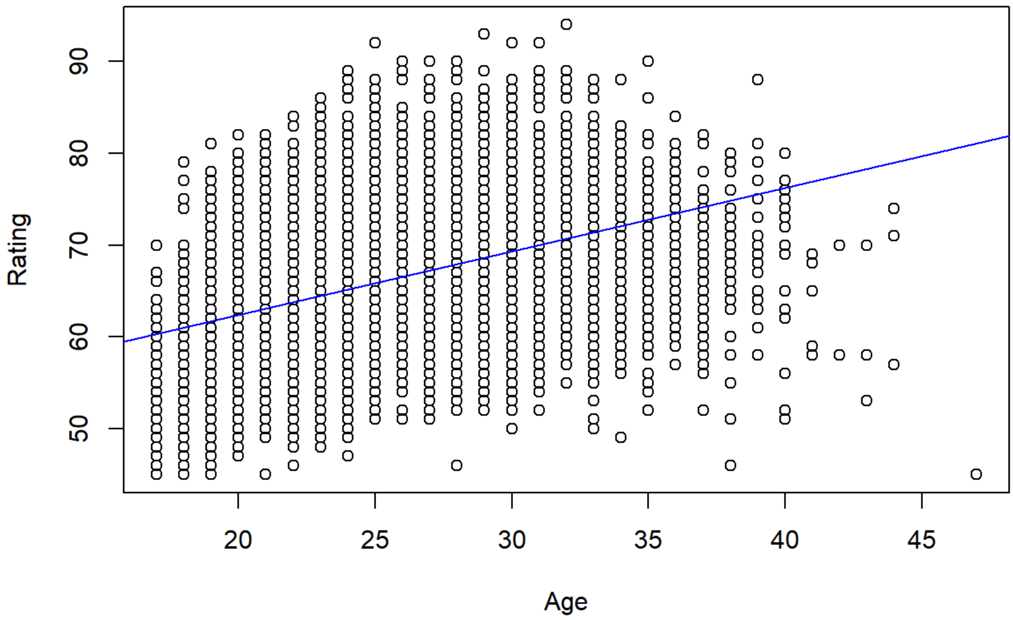
W pierwszym kroku użyto regresji liniowej.

```
ageRatingLinear <- lm(Rating ~ Age, data = Fifa)
summary(ageRatingLinear)
```

```
##
## Call:
## lm(formula = Rating ~ Age, data = Fifa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.105  -4.234  -0.234   3.927  26.153
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.50813    0.26258  184.74   <2e-16 ***
## Age          0.69355    0.01014   68.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.296 on 17586 degrees of freedom
## Multiple R-squared:   0.21,	Adjusted R-squared:   0.21
## F-statistic:  4675 on 1 and 17586 DF,  p-value: < 2.2e-16
```

Prezentacja graficzna dopasowania:

```
plot(Age, Rating)
abline(ageRatingLinear, col='blue')
```



Regresja wielomianowa

Patrząc na powyższy wykres, ale także analizując logicznie zależność ogólnej oceny zawodnika od wieku wydaje się, że do pewnego momentu w karierze zawodnika jego **Rating** rośnie, a następnie powinien spadać.

Z tego względu postanowiono zbadać, czy istnieją oznaki istotnej nieliniowej zależności między **Age** a **Rating**.

Dopasowano model regresji wielomianowej trzeciego stopnia:

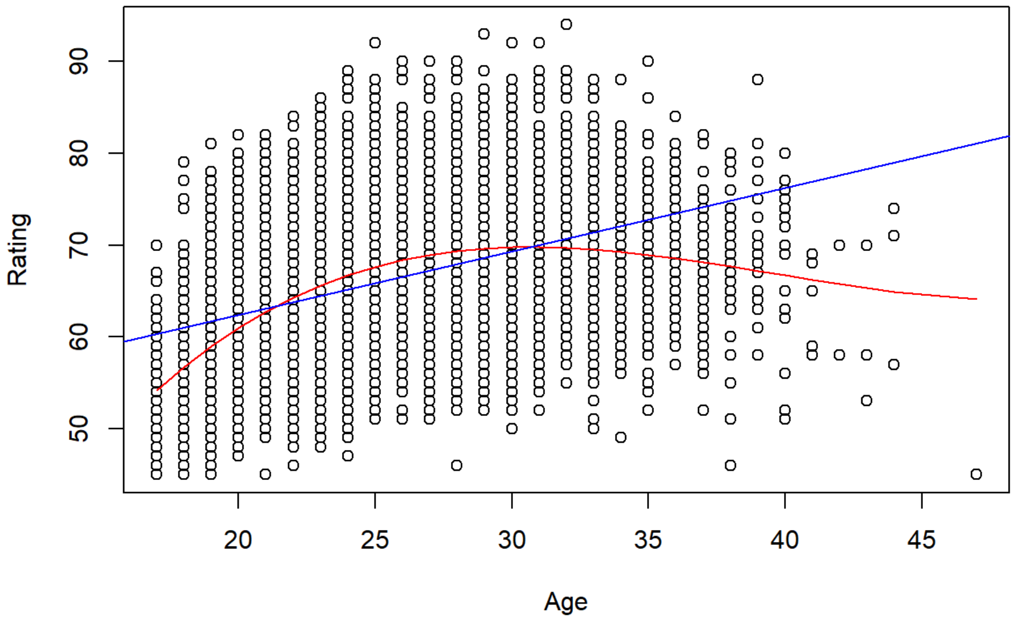
```
ageRatingPoly <- lm(Rating ~ poly(Age, 3), data = Fifa)
summary(ageRatingPoly)
```

```
##
## Call:
## lm(formula = Rating ~ poly(Age, 3), data = Fifa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.3575  -3.9424  -0.3715   3.6425  24.3676
##
## Coefficients:
##                Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    66.16619    0.04501 1470.136  < 2e-16 ***
## poly(Age, 3)1  430.46835    5.96880   72.120  < 2e-16 ***
## poly(Age, 3)2 -261.43378    5.96880  -43.800  < 2e-16 ***
## poly(Age, 3)3   47.07163    5.96880    7.886 3.29e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.969 on 17584 degrees of freedom
## Multiple R-squared:   0.29,  Adjusted R-squared:  0.2899
## F-statistic:  2394 on 3 and 17584 DF,  p-value: < 2.2e-16
```

Porównując dopasowanie w modelu liniowym i wielomianowym można zauważyć m.in. wzrost statystki **Multiple R-squared** z **0.21** do **0.29**. Potwierdza to, że zastosowanie nieliniowego modelu w tym przypadku jest jak najbardziej zasadne.

W celu dodatkowej wizualizacji różnicy w dopasowaniach, oba dopasowania przedstawiono na tle danych na jednym wykresie:

```
plot(Age, Rating)
lines(sort(Age), fitted(ageRatingPoly)[order(Age)], col='red')
abline(ageRatingLinear, col='blue')
```



Które cechy mają największy wpływ na Rating zawodnika?

Na początku poszukujemy optymalnego zbioru cech, który ma największy wpływ na rating zawodnika. W tym celu zastosujemy: * regresję wielokrotną, * selekcja cech w modelach liniowych, * selekcja krokowa do przodu i wstecz * drzewa regresyjne * bagging * lasy losoowe * boosting

Regresja wielokrotna

```
lmFit.many <- lm(Rating ~ ., data = Fifa1h)
summary(lmFit.many)
```

```
##
## Call:
## lm(formula = Rating ~ ., data = Fifa1h)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5724  -1.8074  -0.0015   1.7981  14.2181
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.518e+02  2.588e+01  -5.864 4.60e-09 ***
## Club_Kit                -5.150e-03  1.136e-03  -4.532 5.89e-06 ***
## Contract_Expiry          7.734e-02  1.275e-02   6.067 1.33e-09 ***
## Height                   1.082e-02  6.385e-03   1.695 0.090063 .
## Weight                   2.046e-02  5.388e-03   3.798 0.000146 ***
## Age                      6.683e-02  6.160e-03  10.848  < 2e-16 ***
## Weak_foot                1.178e-01  3.549e-02   3.320 0.000901 ***
## Skill_Moves              1.060e+00  4.947e-02  21.430  < 2e-16 ***
## Ball_Control             1.772e-01  5.065e-03  34.984  < 2e-16 ***
## Dribbling                1.525e-02  4.210e-03   3.623 0.000292 ***
## Marking                  1.052e-02  4.103e-03   2.564 0.010351 *
## Sliding_Tackle          -1.616e-02  4.710e-03  -3.432 0.000601 ***
## Standing_Tackle          4.394e-02  4.904e-03   8.961  < 2e-16 ***
## Aggression               5.575e-03  2.282e-03   2.443 0.014568 *
## Reactions                2.743e-01  3.806e-03  72.082  < 2e-16 ***
## Attacking_Position      -4.176e-02  3.225e-03 -12.950  < 2e-16 ***
## Interceptions            4.944e-03  3.188e-03   1.551 0.120925
## Vision                  -7.900e-03  2.953e-03  -2.675 0.007477 **
## Composure                4.547e-02  2.736e-03  16.618  < 2e-16 ***
## Crossing                 1.633e-02  2.884e-03   5.663 1.51e-08 ***
## Short_Pass               7.251e-02  4.833e-03  15.002  < 2e-16 ***
## Long_Pass               -2.194e-02  3.729e-03  -5.884 4.07e-09 ***
## Acceleration             3.033e-02  4.289e-03   7.072 1.59e-12 ***
## Speed                    3.643e-02  4.104e-03   8.879  < 2e-16 ***
## Stamina                  8.437e-03  2.637e-03   3.199 0.001380 **
## Strength                 3.651e-02  2.954e-03  12.357  < 2e-16 ***
## Balance                 -1.824e-02  3.274e-03  -5.572 2.55e-08 ***
## Agility                  2.425e-03  3.059e-03   0.793 0.427979
## Jumping                  1.599e-02  2.291e-03   6.979 3.08e-12 ***
## Heading                  1.049e-01  2.934e-03  35.746  < 2e-16 ***
## Shot_Power               2.311e-02  3.018e-03   7.657 2.01e-14 ***
## Finishing                2.625e-02  3.457e-03   7.592 3.31e-14 ***
## Long_Shots              -2.413e-02  3.231e-03  -7.470 8.40e-14 ***
## Curve                    1.106e-02  2.970e-03   3.724 0.000197 ***
## Freekick_Accuracy       -4.243e-03  2.657e-03  -1.597 0.110383
## Penalties                4.335e-03  2.848e-03   1.522 0.128041
## Volleys                 -7.209e-03  2.978e-03  -2.420 0.015515 *
## GK_Positioning           8.726e-02  6.116e-03  14.268  < 2e-16 ***
## GK_Diving                7.027e-02  6.139e-03  11.447  < 2e-16 ***
## GK_Kicking               3.523e-02  5.724e-03   6.154 7.70e-10 ***
## GK_Handling              8.852e-02  6.174e-03  14.338  < 2e-16 ***
## GK_Reflexes              8.788e-02  6.116e-03  14.368  < 2e-16 ***
## Nationality_encoded     -9.848e-05  4.464e-04  -0.221 0.825413
## Club_encoded             1.766e-04  1.153e-04   1.532 0.125624
## Preffered_Foot_encoded  -2.040e-01  5.253e-02  -3.884 0.000103 ***
## Preffered_Position_encoded -3.406e-03  3.856e-04  -8.831  < 2e-16 ***
## Work_Rate_encoded       -6.620e-02  8.374e-03  -7.905 2.84e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.78 on 17540 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8463, Adjusted R-squared:  0.8459
## F-statistic:  2099 on 46 and 17540 DF,  p-value: < 2.2e-16
```

Regresja wielokrotna wskazuję, że cechami na które warto zwrócić by uwagę w dalej analizie są m.in: * Age * Skill_Moves * Ball_Control * Standing_Tackle * Reactions * Attacking_Position * Composure * Short_Pass * Speed * Strength * Heading * GK_Positioning
* GK_Diving
* GK_Kicking
* GK_Handling
* GK_Reflexes

* Preffered_Position

```
Fifa_selected <- select(Fifa1h, Rating, Age, Skill_Moves, Ball_Control, Standing_Tackle, Reactions, Attac
king_Position, Composure, Short_Pass, Speed, Strength, Heading, GK_Positioning, GK_Diving, GK_Kicking, GK
_Handling, GK_Reflexes, Preffered_Position_encoded)
```

```
lmFit <- lm(Rating ~ ., data = Fifa_selected)
summary(lmFit)
```

```
##
## Call:
## lm(formula = Rating ~ ., data = Fifa_selected)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.732  -1.845   0.015   1.823  15.649
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.0429953  0.2542253  23.770  < 2e-16 ***
## Age                        0.0704165  0.0056962  12.362  < 2e-16 ***
## Skill_Moves                1.1480812  0.0481632  23.837  < 2e-16 ***
## Ball_Control               0.1951679  0.0044282  44.074  < 2e-16 ***
## Standing_Tackle            0.0410400  0.0017874  22.961  < 2e-16 ***
## Reactions                  0.2821578  0.0037522  75.198  < 2e-16 ***
## Attacking_Position        -0.0273090  0.0026471 -10.316  < 2e-16 ***
## Composure                  0.0475462  0.0026635  17.851  < 2e-16 ***
## Short_Pass                 0.0579150  0.0039103  14.811  < 2e-16 ***
## Speed                      0.0723433  0.0023129  31.278  < 2e-16 ***
## Strength                   0.0548486  0.0022668  24.196  < 2e-16 ***
## Heading                    0.1170531  0.0025643  45.648  < 2e-16 ***
## GK_Positioning             0.0847428  0.0062072  13.652  < 2e-16 ***
## GK_Diving                  0.0702003  0.0062350  11.259  < 2e-16 ***
## GK_Kicking                 0.0343488  0.0058070   5.915 3.38e-09 ***
## GK_Handling                0.0891626  0.0062591  14.245  < 2e-16 ***
## GK_Reflexes                0.0884226  0.0062057  14.249  < 2e-16 ***
## Preffered_Position_encoded -0.0018921  0.0003636  -5.204 1.97e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.828 on 17570 degrees of freedom
## Multiple R-squared:  0.8408, Adjusted R-squared:  0.8406
## F-statistic:  5458 on 17 and 17570 DF,  p-value: < 2.2e-16
```

Z powyższej krótkiej analizy regresji wielokrotnej wynika, że najważniejszym współczynnikami z punktu widzenia Ratingu zawodnika są:

- Reactions
- Heading
- Ball_Control
- Speed
- Strength
- Standing_Tackle

Selekcja cech modelu liniowego

```
fit.bs <- regsubsets(Rating ~ ., data = Fifa1h[1:2000,], nvmax = 47)
fit.bs.summary <- summary(fit.bs)
fit.bs.summary
```

Obiekt zwracany przez funkcję summary.regsubsets() zawiera informacje umożliwiające zidentyfikowanie globalnie najlepszego pozdbioru cech, np. miarę Cp.

```
fit.bs.summary$cp
```

```
##  [1] 652.62365 595.48095 501.36443 419.75832 335.94043 260.33939 211.66962
##  [8] 158.67852 135.51454 111.78142  79.21138  71.19793  59.16293  51.16492
## [15]  44.64462  38.36763  32.46693  27.77737  23.23904  17.90119  17.60798
## [22]  16.63355  16.23296  16.72903  16.23323  16.09142  16.30181  17.03525
## [29]  18.02624  19.25321  20.56639  21.78975  23.28500  24.76098  26.27870
## [36]  27.86808  29.48297  31.21764  33.07786  35.03447  37.00894  39.00187
## [43]  41.00014  43.00002  45.00000  47.00000
```

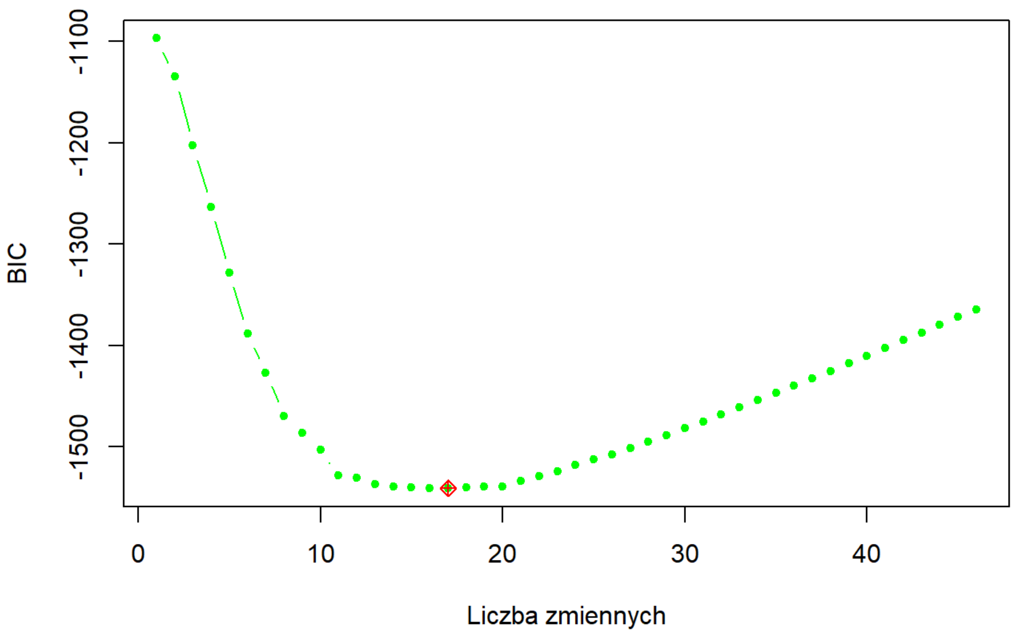
```
bic.min <- which.min(fit.bs.summary$bic)
bic.min
```

```
## [1] 17
```

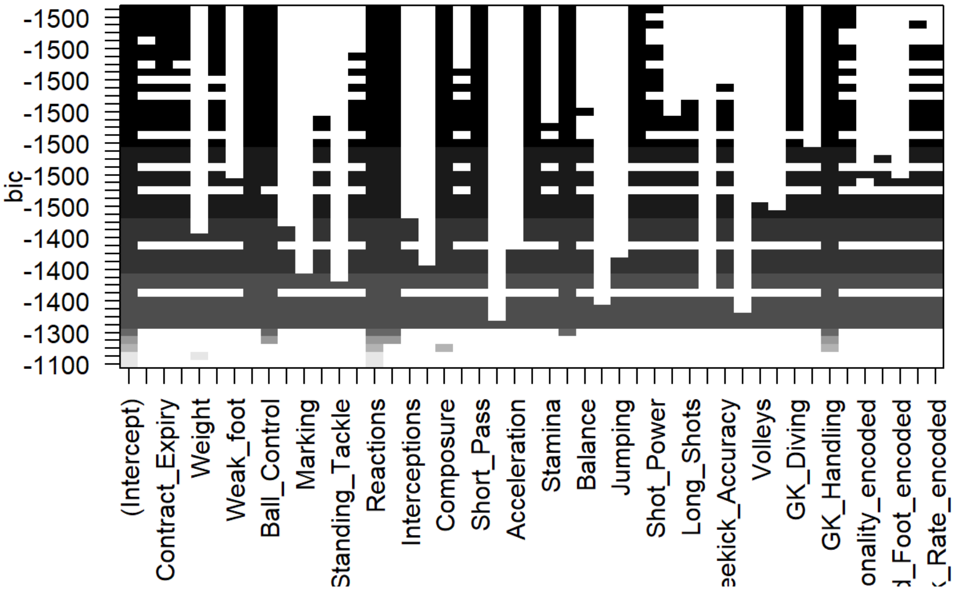
```
fit.bs.summary$bic[bic.min]
```

```
## [1] -1540.892
```

```
plot(fit.bs.summary$bic, xlab = "Liczba zmiennych", ylab = "BIC", col = "green",
     type = "b", pch = 20)
points(bic.min, fit.bs.summary$bic[bic.min], col = "red", pch = 9)
```




```
plot(fit.bs, scale = "bic")
```




```r
coef(fit.bs, id = 17)
```

```
##       (Intercept)          Club_Kit   Contract_Expiry            Height
##     -296.55706133       -0.00773203        0.15955240        0.03600057
##               Age       Skill_Moves      Ball_Control         Reactions
##        0.06634768        0.78214993        0.07266140        0.31591879
## Attacking_Position         Composure        Short_Pass             Speed
##       -0.06956778        0.03485960        0.07522672        0.04135903
##          Strength           Heading        Shot_Power         GK_Diving
##        0.02417339        0.02927317        0.01613018        0.04666119
##       GK_Handling       GK_Reflexes
##        0.05744045        0.03615208
```

```r
fit.forward <- regsubsets(Rating ~ ., data = Fifa1h[1:nrow(Fifa1h)/2,], nvmax = 47, method = "forward")
fit.forward.summary <- summary(fit.forward)
fit.forward.summary
```

```r
fit.backward <- regsubsets(Rating ~ ., data = Fifa1h[1:nrow(Fifa1h)/2,], nvmax = 47, method = "backward")
fit.backward.summary <- summary(fit.backward)
fit.backward.summary
```

```r
fit.backward.summary$cp
```

```
##  [1] 9873.72791 9407.14484 6642.81347 4751.56254 4083.76792 3442.77585
##  [7] 2811.50631 2250.80508 1740.79643 1434.37278 1070.36141  810.78679
## [13]  668.80060  528.65943  459.81921  398.03641  377.43724  317.61038
## [19]  266.32306  223.29302  199.59962  174.00645  153.90822  136.53559
## [25]  118.63640  102.12615   90.38224   75.87439   66.03159   57.14609
## [31]   51.14415   47.54842   43.89387   43.30747   40.85303   40.39099
## [37]   39.63987   38.95124   39.49811   39.54629   40.18196   41.25333
## [43]   42.43968   43.90285   45.43601   47.00000
```

```r
fit.forward.summary$cp
```

```
##  [1] 9873.72791 9084.11770 8049.13945 5911.19085 4241.29130 3601.96447
##  [7] 3058.41504 2744.42337 1740.79643 1434.37278 1070.36141  810.78679
## [13]  668.80060  528.65943  459.81921  398.03641  348.58264  300.80469
## [19]  268.47782  240.55578  199.59962  174.00645  153.90822  136.53559
## [25]  118.63640  102.12615   90.38224   75.87439   66.03159   57.14609
## [31]   51.14415   47.54842   43.89387   43.30747   40.85303   40.21875
## [37]   39.79397   38.95124   39.49811   39.54629   40.18196   41.25333
## [43]   42.43968   43.90285   45.43601   47.00000
```

```
bic.min <- which.min(fit.backward.summary$bic)
bic.min
```

```
## [1] 30
```

```
fit.backward.summary$bic[bic.min]
```

```
## [1] -20099.49
```

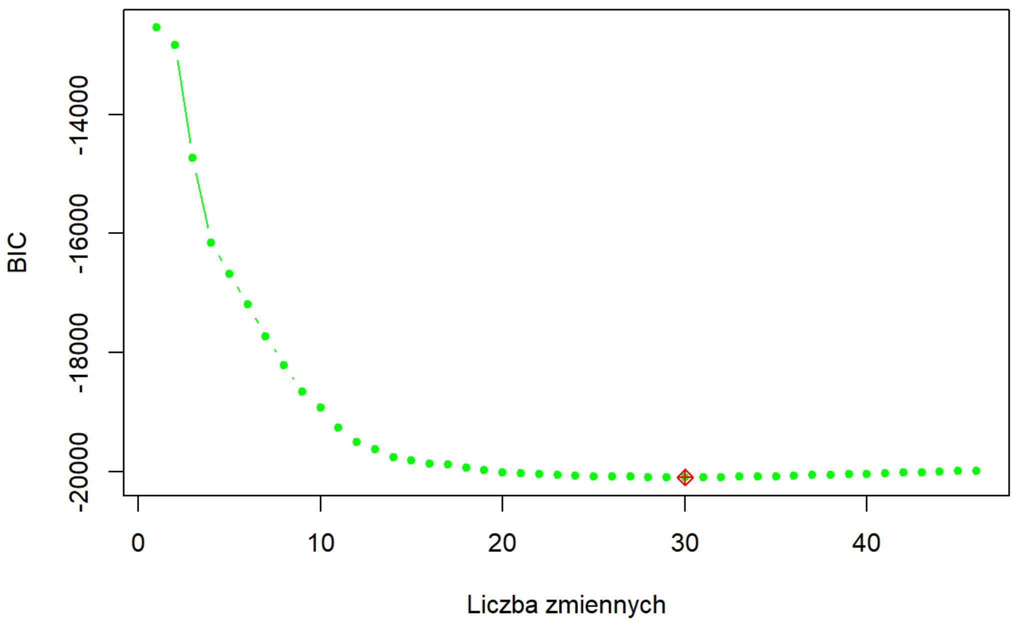
```
bic.min <- which.min(fit.forward.summary$bic)
bic.min
```

```
## [1] 30
```

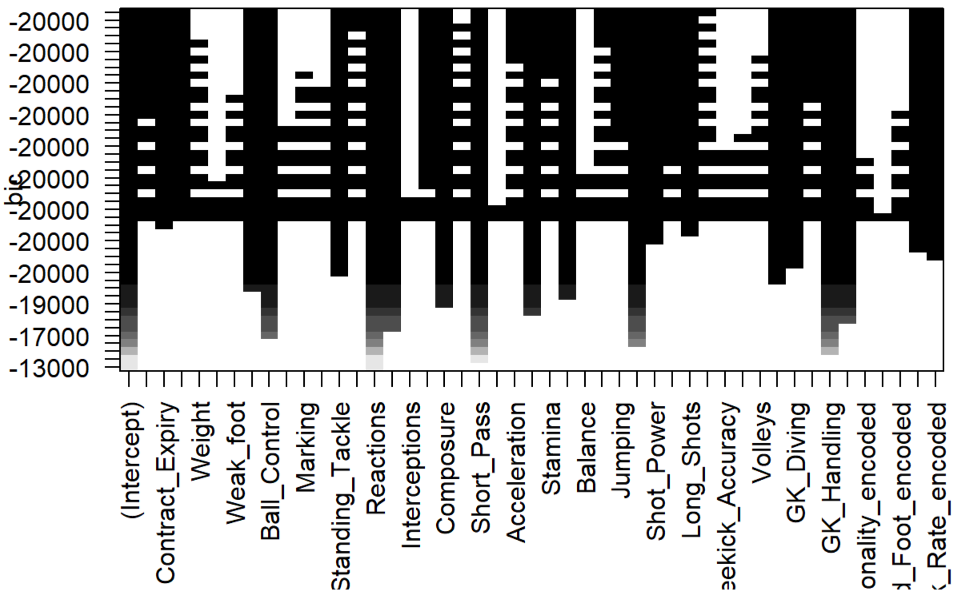
```
fit.forward.summary$bic[bic.min]
```

```
## [1] -20099.49
```

```
plot(fit.backward.summary$bic, xlab = "Liczba zmiennych", ylab = "BIC", col = "green",
     type = "b", pch = 20)
points(bic.min, fit.backward.summary$bic[bic.min], col = "red", pch = 9)
```




```
plot(fit.backward, scale = "bic")
```




```
coef(fit.backward, id = 30)
```

```
##             (Intercept)                Club_Kit
##           -1.787279e+02           -4.509855e-03
##         Contract_Expiry                  Height
##            9.642471e-02            3.227771e-02
##             Skill_Moves            Ball_Control
##            8.101838e-01            1.371137e-01
##         Standing_Tackle              Aggression
##            1.592046e-02            6.723965e-03
##               Reactions      Attacking_Position
##            2.801764e-01           -4.655247e-02
##                  Vision               Composure
##           -1.382535e-02            3.871448e-02
##              Short_Pass            Acceleration
##            1.010068e-01            1.783863e-02
##                   Speed                 Stamina
##            3.783483e-02           -1.030202e-02
##                Strength                 Agility
##            2.438192e-02           -1.165206e-02
##                 Jumping                 Heading
##            9.725594e-03            7.031062e-02
##              Shot_Power               Finishing
##            1.794015e-02            1.485191e-02
##              Long_Shots                   Curve
##           -2.040893e-02            6.660428e-03
##          GK_Positioning               GK_Diving
##            6.341766e-02            5.976209e-02
##              GK_Kicking             GK_Handling
##            2.097294e-02            6.551359e-02
##             GK_Reflexes Preffered_Position_encoded
##            5.967447e-02           -2.886439e-03
##       Work_Rate_encoded
##           -6.270920e-02
```

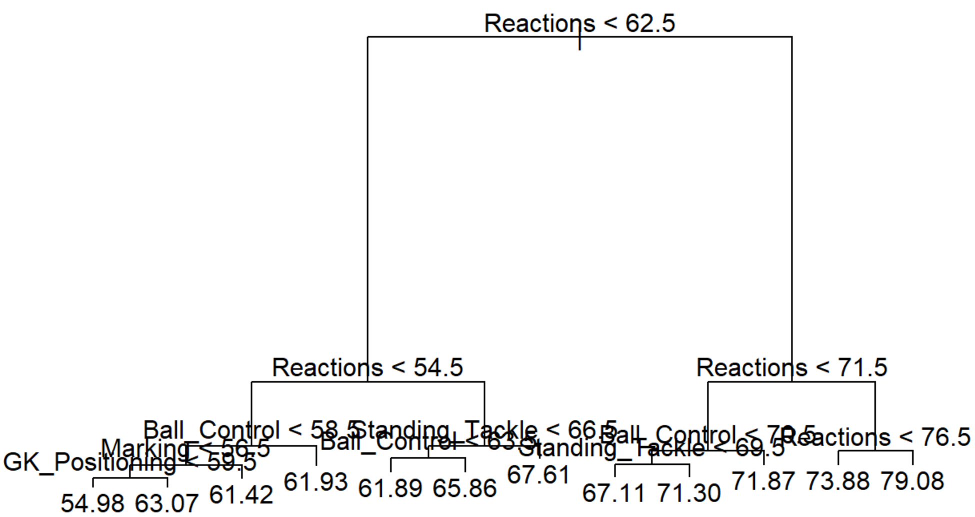
Drzewa decyzyjne

Drzewa regresyjne

```
rating.tree <- tree(Rating ~ ., data = Fifa1h)
summary(rating.tree)
```

```
##
## Regression tree:
## tree(formula = Rating ~ ., data = Fifa1h)
## Variables actually used in tree construction:
## [1] "Reactions"       "Ball_Control"    "Marking"          "GK_Positioning"
## [5] "Standing_Tackle"
## Number of terminal nodes:  12
## Residual mean deviance:  12.07 = 212200 / 17580
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -20.0800  -2.0780   0.1119   0.0000   2.1180  17.5800
```

```
plot(rating.tree)
text(rating.tree, pretty = 0)
```



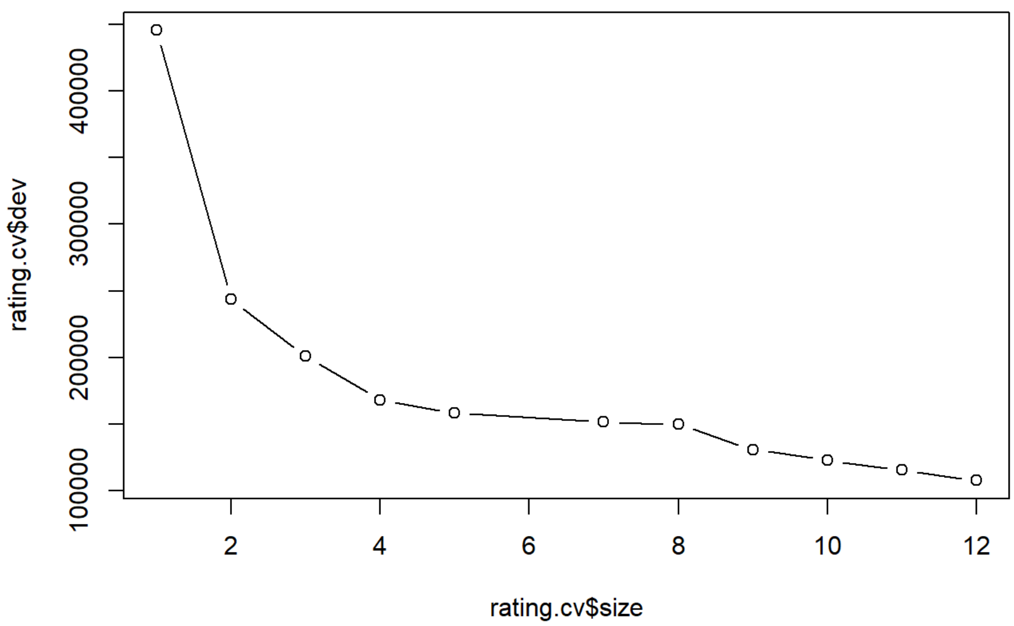

```
rating.tree
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 17587 882100 66.17
##    2) Reactions < 62.5 9088 259900 61.58
##      4) Reactions < 54.5 3932  97070 58.32
##        8) Ball_Control < 58.5 2756  60850 56.78
##         16) Marking < 56.5 2204  39130 55.62
##           32) GK_Positioning < 59.5 2030  26480 54.98 *
##           33) GK_Positioning > 59.5 174   2146 63.07 *
##         17) Marking > 56.5 552   6843 61.42 *
##        9) Ball_Control > 58.5 1176  14400 61.93 *
##      5) Reactions > 54.5 5156  89210 64.07
##       10) Standing_Tackle < 66.5 4259  66770 63.32
##         20) Ball_Control < 63.5 2725  37710 61.89 *
##         21) Ball_Control > 63.5 1534  13570 65.86 *
##       11) Standing_Tackle > 66.5 897   8783 67.61 *
##    3) Reactions > 62.5 8499 226800 71.07
##      6) Reactions < 71.5 5866  87740 69.02
##       12) Ball_Control < 70.5 4436  61580 68.11
##         24) Standing_Tackle < 69.5 3379  38680 67.11 *
##         25) Standing_Tackle > 69.5 1057   8773 71.30 *
##       13) Ball_Control > 70.5 1430  10860 71.87 *
##      7) Reactions > 71.5 2633  59780 75.63
##       14) Reactions < 76.5 1749  25140 73.88 *
##       15) Reactions > 76.5 884  18780 79.08 *
```

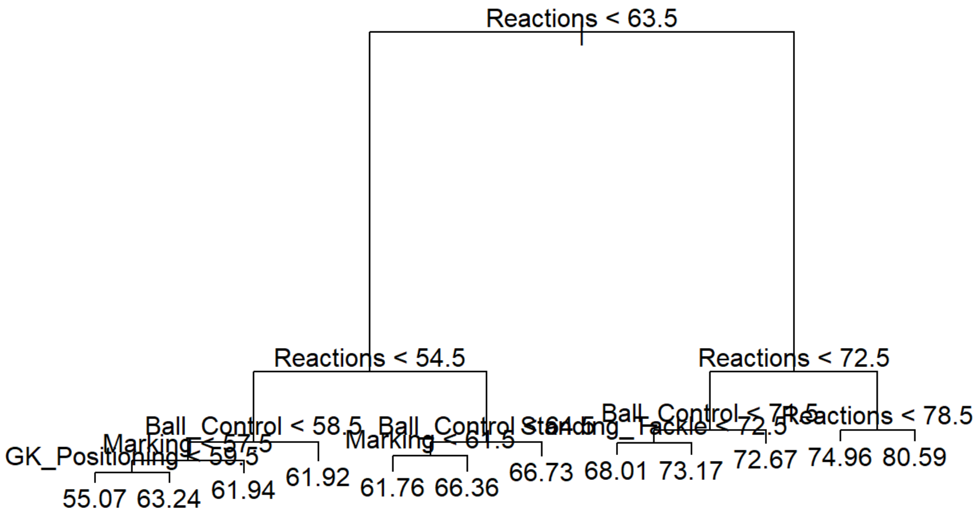
```
set.seed(1)
n <- nrow(Fifa1h)
train <- sample(1:n, n / 2)
test <- -train
rating.tree <- tree(Rating ~ ., data = Fifa1h, subset = train)
rating.pred <- predict(rating.tree, newdata = Fifa1h[test,])
mean((rating.pred - Fifa1h$Rating[test])^2)
```

```
## [1] 12.26272
```

```
rating.cv <- cv.tree(rating.tree)
plot(rating.cv$size, rating.cv$dev, type = "b")
```



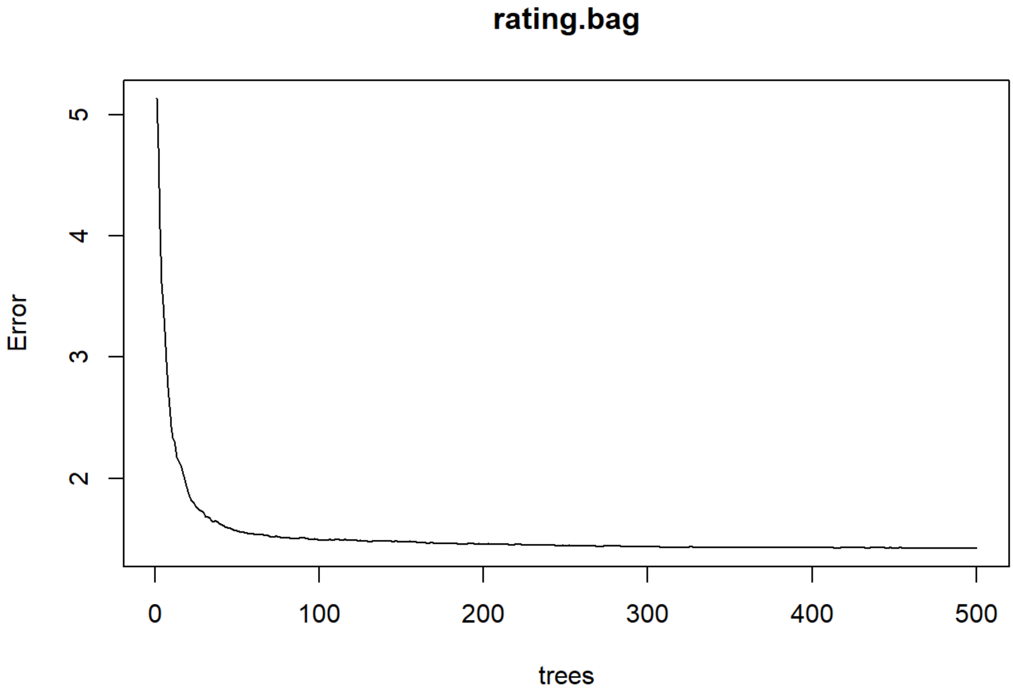
```
rating.pruned <- prune.tree(rating.tree, best = 12)
plot(rating.pruned)
text(rating.pruned)
```




```
rating.bag <- randomForest(Rating ~ ., data = Fifa1h[1:2000,], mtry = 10, importance = TRUE, na.action=na
.roughfix)
rating.bag
```

```
##
## Call:
##  randomForest(formula = Rating ~ ., data = Fifa1h[1:2000, ], mtry = 10,      importance = TRUE, na.act
ion = na.roughfix)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 1.423207
##                     % Var explained: 85.17
```

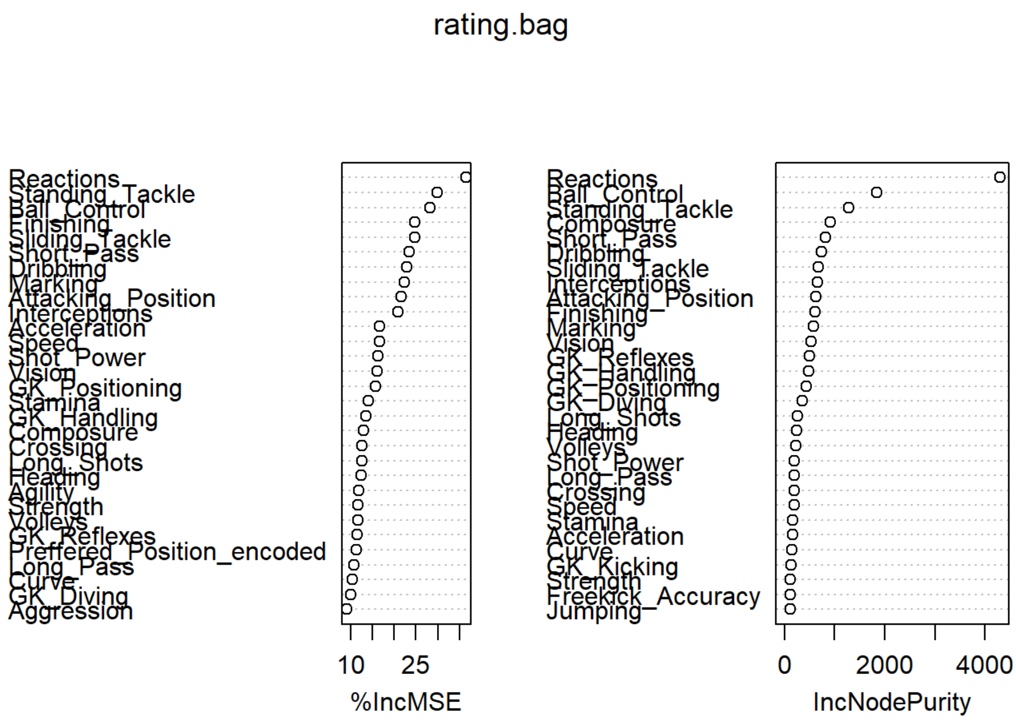
```
plot(rating.bag, type = "l")
```




```
importance(rating.bag)
```

```
##                          %IncMSE IncNodePurity
## Club_Kit                 3.106361     92.008486
## Contract_Expiry          2.824361     50.413598
## Height                   5.771448     81.293125
## Weight                   5.238821     87.432114
## Age                      2.895244     75.358909
## Weak_foot                1.376934     23.187692
## Skill_Moves              5.365105     32.163508
## Ball_Control            28.220104   1848.476240
## Dribbling               22.925777    740.629169
## Marking                 22.399133    573.274531
## Sliding_Tackle          24.719959    673.394638
## Standing_Tackle         29.969612   1282.161754
## Aggression               9.002118    114.897187
## Reactions               36.576773   4306.469541
## Attacking_Position      21.641150    627.816317
## Interceptions           20.873588    660.655053
## Vision                  16.028836    537.851981
## Composure               12.823357    922.544614
## Crossing                12.592665    191.682786
## Short_Pass              23.528171    827.432250
## Long_Pass               10.741949    195.380445
## Acceleration            16.579813    160.042836
## Speed                   16.561842    189.492802
## Stamina                 13.950592    161.841887
## Strength                11.594624    124.407917
## Balance                  8.049026     89.352127
## Agility                 11.859445    112.795024
## Jumping                  5.937057    115.788461
## Heading                 12.434539    241.702364
## Shot_Power              16.167282    196.283560
## Finishing               24.754539    604.044564
## Long_Shots              12.519810    256.375535
## Curve                   10.386941    147.746041
## Freekick_Accuracy        5.054647    118.754362
## Penalties                4.410202    110.620478
## Volleys                 11.572699    236.097996
## GK_Positioning          15.616670    429.022261
## GK_Diving               10.003104    349.195628
## GK_Kicking               5.250074    132.987402
## GK_Handling             13.542031    490.366308
## GK_Reflexes             11.487956    507.419025
## Nationality_encoded      1.413140     71.211723
## Club_encoded             3.144618    101.025863
## Preffered_Foot_encoded  -1.136055      9.883755
## Preffered_Position_encoded 11.310371     85.981085
## Work_Rate_encoded        3.184776     45.006065
```

```
varImpPlot(rating.bag)
```

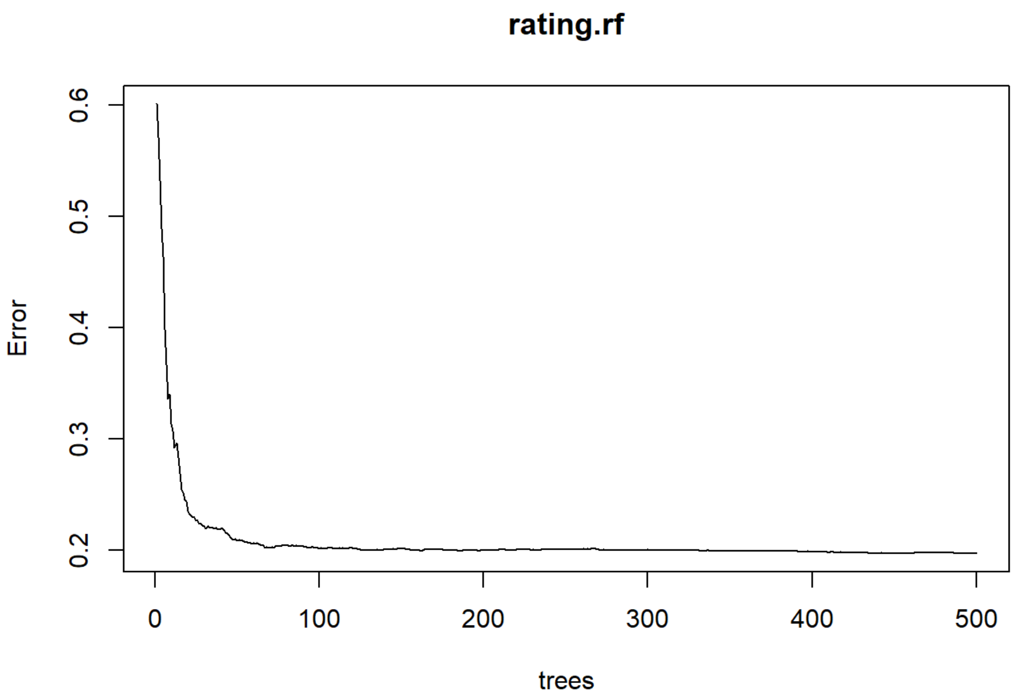


Lasy losoowe

```r
set.seed(2)
rating.rf <- randomForest(Rating ~ ., data = Fifa1h[1:2000,], subset = train,
                          importance = TRUE, na.action=na.roughfix)
rating.pred.rf <- predict(rating.rf, newdata = Fifa1h[test,])
mean((rating.pred.rf - Fifa1h$Rating[test])^2)
```

```
## [1] 135.0409
```

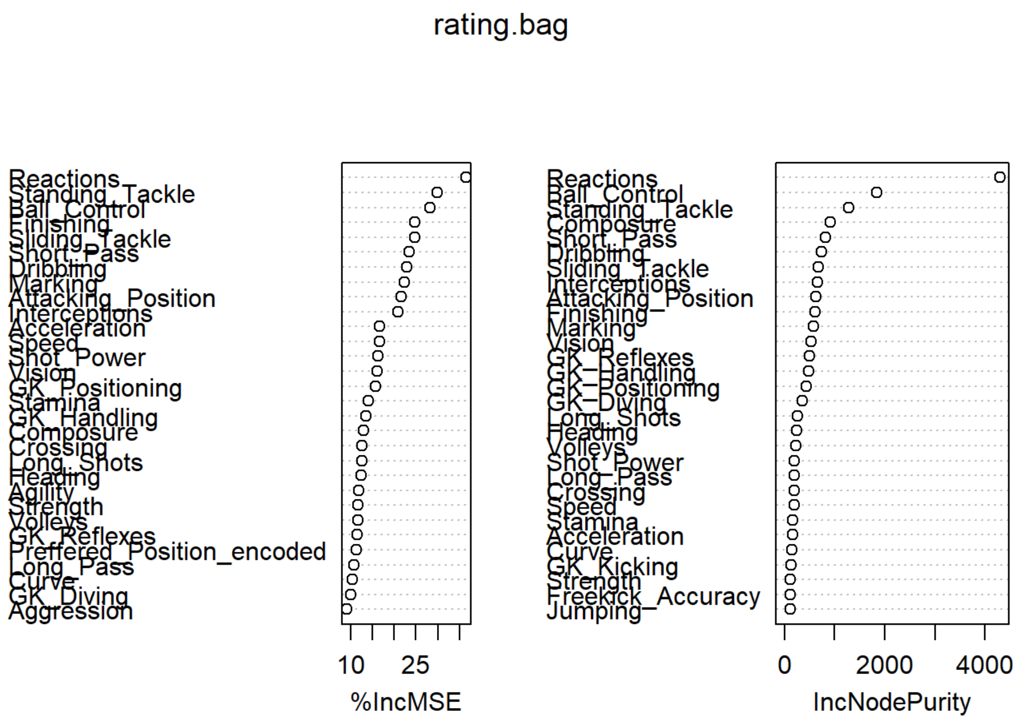
```r
plot(rating.rf, type = "l")
```




```r
importance(rating.rf)
```

```
##                             %IncMSE IncNodePurity
## Club_Kit                    9.237220     67.965688
## Contract_Expiry             6.292984     29.813992
## Height                      7.517606     37.337123
## Weight                      8.591278     52.565606
## Age                         4.347809     38.908536
## Weak_foot                   2.544246     12.326419
## Skill_Moves                 4.460878     12.035900
## Ball_Control               29.239745   1149.054354
## Dribbling                  16.626497    298.504650
## Marking                    16.890611    300.940297
## Sliding_Tackle             20.553267    401.300073
## Standing_Tackle            25.669700    817.696842
## Aggression                  9.695072     77.904515
## Reactions                  40.287441   2786.844066
## Attacking_Position         14.779495    276.265725
## Interceptions              14.409828    267.001498
## Vision                     11.073682    186.447829
## Composure                  11.232387    819.453451
## Crossing                    9.864109     86.887540
## Short_Pass                 19.411709    611.238525
## Long_Pass                   8.862398     74.124698
## Acceleration               11.291445     75.683147
## Speed                      10.377590     80.713457
## Stamina                    10.140981     70.849897
## Strength                   11.988328     63.549604
## Balance                     8.752502     43.200755
## Agility                     7.141115     49.581009
## Jumping                    10.902827     84.861109
## Heading                     9.377662     90.863560
## Shot_Power                  8.981559     94.441320
## Finishing                  16.423983    293.281036
## Long_Shots                  9.543369     88.256971
## Curve                       6.493249     68.130003
## Freekick_Accuracy           7.403186     64.846800
## Penalties                   7.915907     50.472551
## Volleys                     9.307049     97.586616
## GK_Positioning             11.967984    132.897208
## GK_Diving                  14.143081    152.348842
## GK_Kicking                  6.805156     59.806866
## GK_Handling                13.623396    187.277116
## GK_Reflexes                13.365042    210.080832
## Nationality_encoded         5.897881     45.024915
## Club_encoded                7.318435     58.490042
## Preffered_Foot_encoded      2.965265      5.555083
## Preffered_Position_encoded  8.823785     42.814402
## Work_Rate_encoded           4.649064     22.598128
```

```
varImpPlot(rating.bag)
```

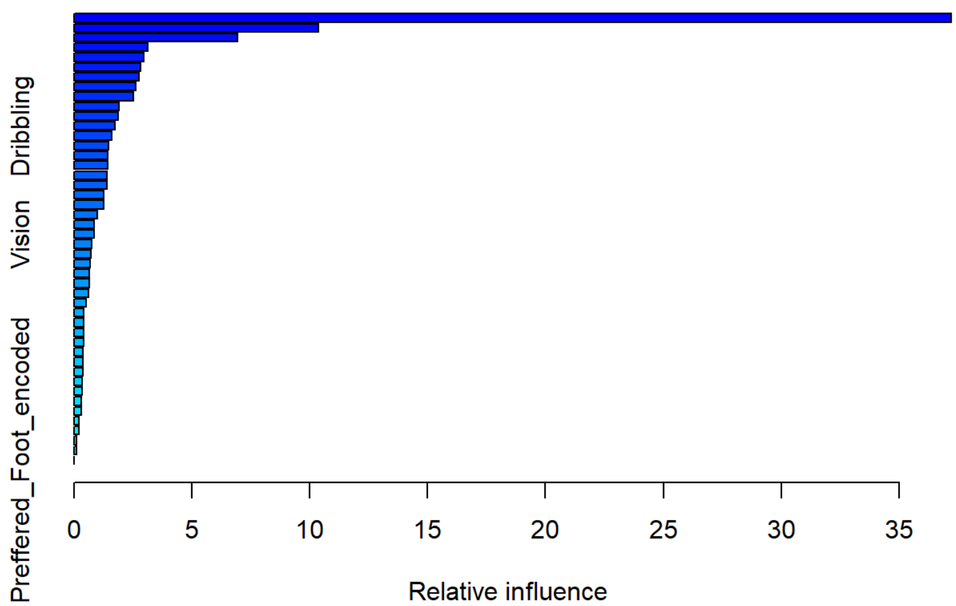


Boosting

```r
rating.boost <- gbm(Rating ~ ., data = Fifa1h[1:2000,], distribution = "gaussian",
                    n.trees = 5000, interaction.depth = 4)
rating.boost
```

```
## gbm(formula = Rating ~ ., distribution = "gaussian", data = Fifa1h[1:2000,
##     ], n.trees = 5000, interaction.depth = 4)
## A gradient boosted model with gaussian loss function.
## 5000 iterations were performed.
## There were 46 predictors of which 46 had non-zero influence.
```

```r
summary(rating.boost)
```



```
##                                                     var     rel.inf
## Reactions                                     Reactions 37.21261276
## Ball_Control                               Ball_Control 10.38008476
## Standing_Tackle                         Standing_Tackle  6.92507694
## GK_Handling                                 GK_Handling  3.13655677
## Short_Pass                                   Short_Pass  2.95111645
## Sliding_Tackle                           Sliding_Tackle  2.81232260
## Finishing                                     Finishing  2.74856780
## GK_Positioning                           GK_Positioning  2.63755241
## GK_Reflexes                                 GK_Reflexes  2.51363840
## Interceptions                             Interceptions  1.90419153
## Attacking_Position                   Attacking_Position  1.87149093
## Dribbling                                     Dribbling  1.73111968
## Marking                                         Marking  1.61109905
## Shot_Power                                   Shot_Power  1.46170048
## GK_Diving                                     GK_Diving  1.44732623
## Composure                                     Composure  1.44130688
## Speed                                             Speed  1.38804025
## Stamina                                         Stamina  1.38100497
## Heading                                         Heading  1.27513111
## Crossing                                       Crossing  1.24825963
## Jumping                                         Jumping  0.97941198
## Strength                                       Strength  0.83836958
## Vision                                           Vision  0.83705529
## Acceleration                               Acceleration  0.75025299
## Preffered_Position_encoded Preffered_Position_encoded  0.71619021
## Club_encoded                               Club_encoded  0.69478746
## Long_Pass                                     Long_Pass  0.64939742
## Long_Shots                                   Long_Shots  0.64209022
## Aggression                                   Aggression  0.60797853
## GK_Kicking                                   GK_Kicking  0.52970559
## Freekick_Accuracy                     Freekick_Accuracy  0.42291147
## Agility                                         Agility  0.42124136
## Penalties                                     Penalties  0.40054008
## Height                                           Height  0.39725881
## Club_Kit                                       Club_Kit  0.38575059
## Weight                                           Weight  0.37782708
## Volleys                                         Volleys  0.36388491
## Nationality_encoded                 Nationality_encoded  0.33827999
## Balance                                         Balance  0.32835550
## Age                                                 Age  0.32555392
## Curve                                             Curve  0.30731076
## Contract_Expiry                         Contract_Expiry  0.20921847
## Work_Rate_encoded                     Work_Rate_encoded  0.19220728
## Skill_Moves                                 Skill_Moves  0.10116344
## Weak_foot                                     Weak_foot  0.09299013
## Preffered_Foot_encoded           Preffered_Foot_encoded  0.01206729
```

Podsumowanie najlepszego doboru cech:

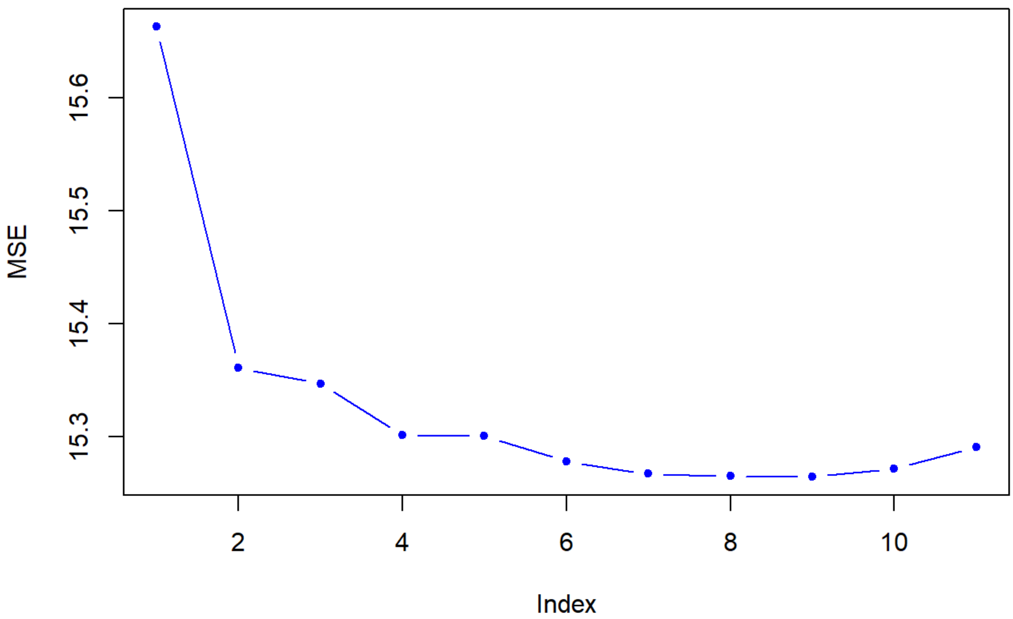
Nazwa metody doboru cech	**Najważniejsze cechy**
Regresja wielokrotna	Reactions, Heading, Ball_Control
Selekcja cech modelu liniowego	Skill_Moves, Reactions, Contract_Expiry
Selekcja krokowa do przodu i wstecz	Skill_Move, Reactions, Ball_Control
Drzewa regresyjne	Reactions, Ball_Control, Marking
Bagging	Reactions, Standing_Tackle, Ball_Control
Lasy losowe	Reactions, Ball_Control, Standing_Tackle
Boosting	Reactions, Ball_Control, Standing_Tackle

Walidacja krzyżowa wybranych cech

```r
set.seed(2)
validation.set <- Fifa1h[-train,]
max.degree <- 11
mse <- rep(0, times = max.degree)
for (i in 1:max.degree) {
  fit.lm <- lm(Rating ~ poly(Reactions, degree = i), data = Fifa1h, subset = train)
  mse[i] <- mean((validation.set$Rating - predict(fit.lm, validation.set))^2)
}
mse
```

```
##  [1] 15.66316 15.36067 15.34681 15.30105 15.30017 15.27759 15.26670 15.26459
##  [9] 15.26400 15.27116 15.29010
```

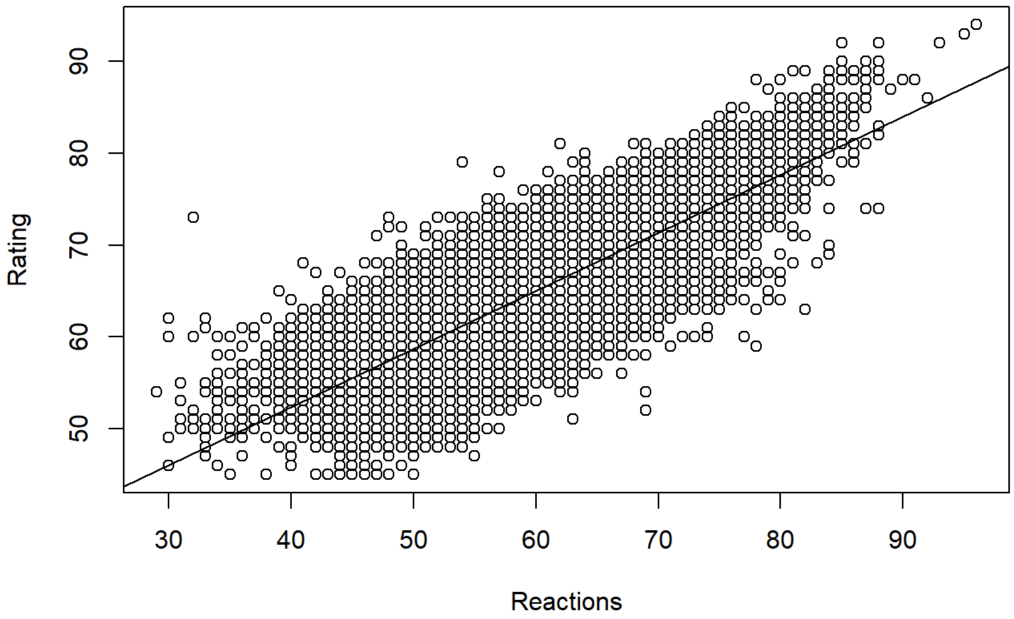
```r
plot(mse, ylab = "MSE", type = "b", pch = 20, col = "blue")
```



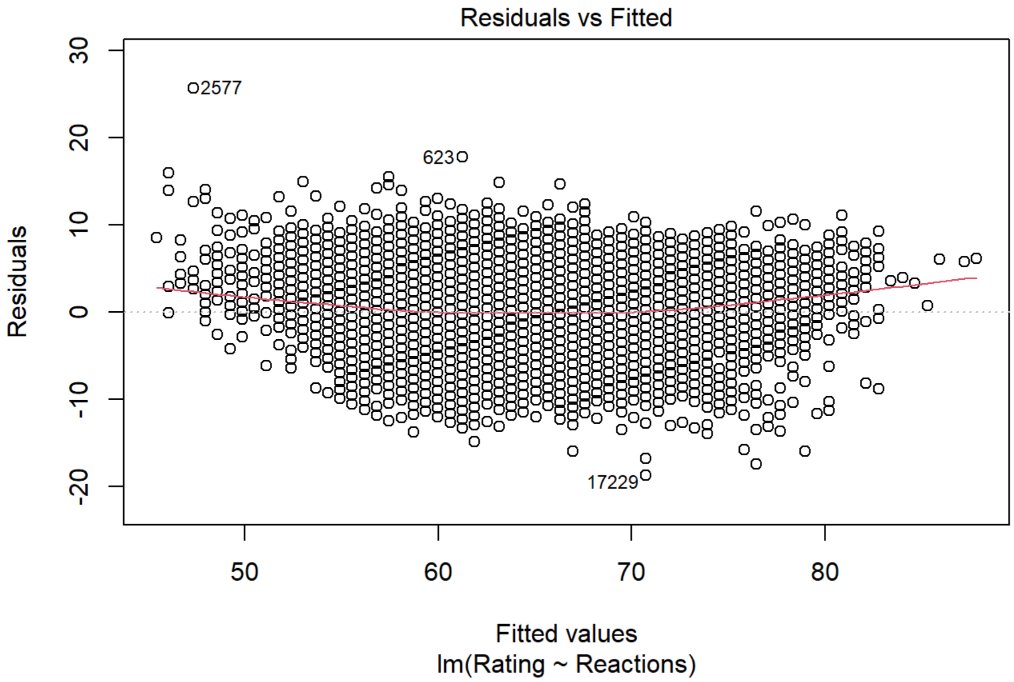

```r
lmFitSimple <- lm(Rating ~ Reactions, data = Fifa1h)
summary(lmFitSimple)
```

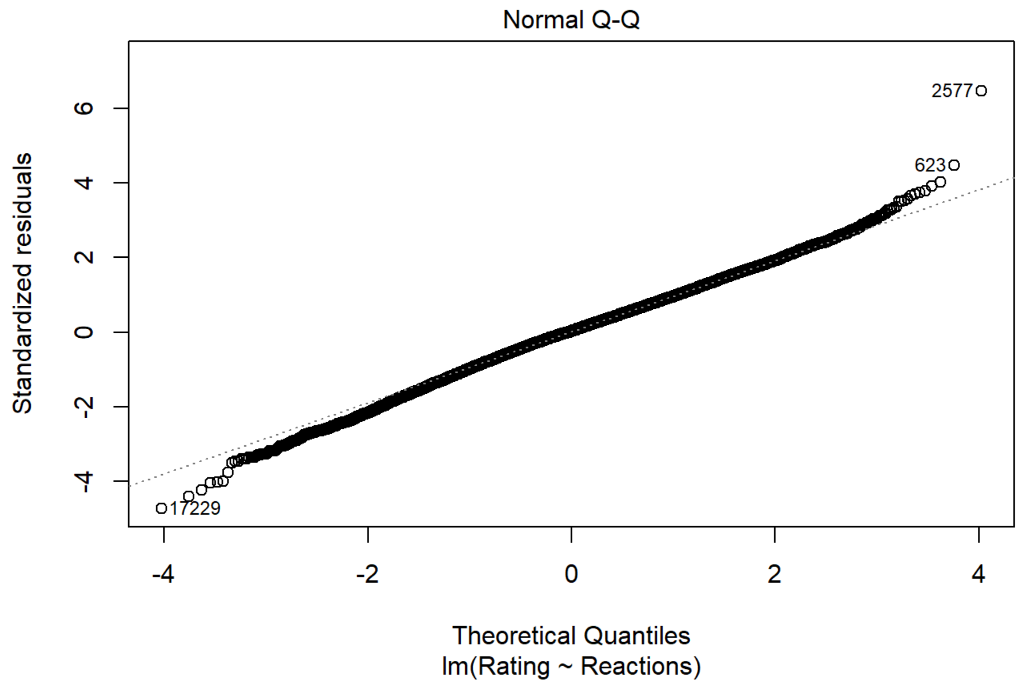
```
## 
## Call:
## lm(formula = Rating ~ Reactions, data = Fifa1h)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -18.7394  -2.4742   0.1168   2.6055  25.6662 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 27.091145   0.201495   134.5   <2e-16 ***
## Reactions    0.632583   0.003226   196.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.968 on 17586 degrees of freedom
## Multiple R-squared:  0.6862, Adjusted R-squared:  0.6862 
## F-statistic: 3.846e+04 on 1 and 17586 DF,  p-value: < 2.2e-16
```

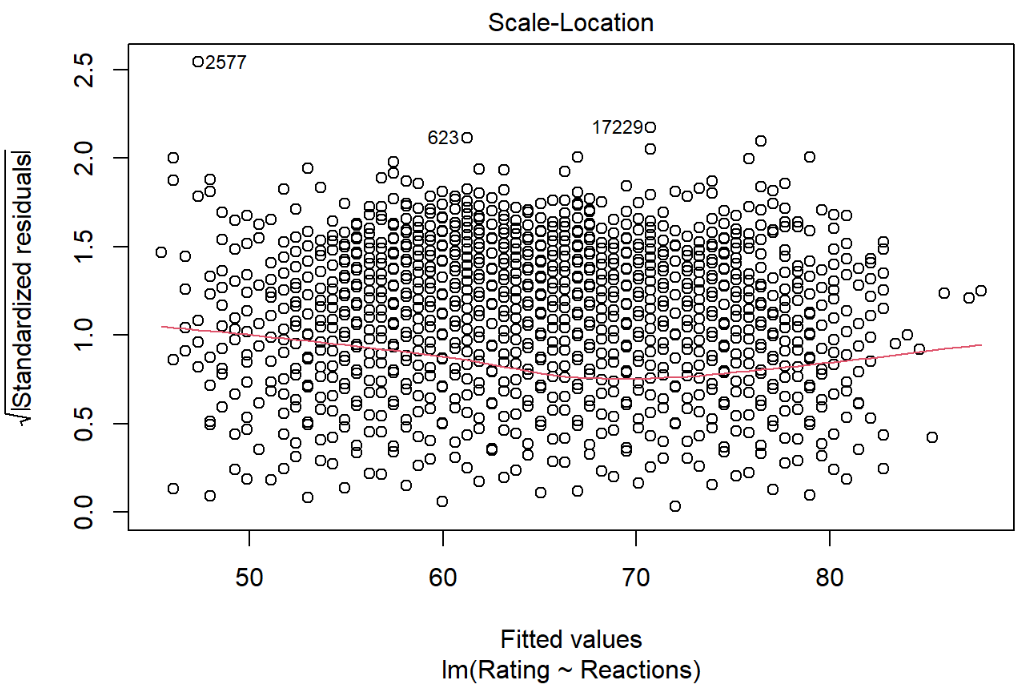
```r
plot(Reactions, Rating)
abline(lmFitSimple)
```

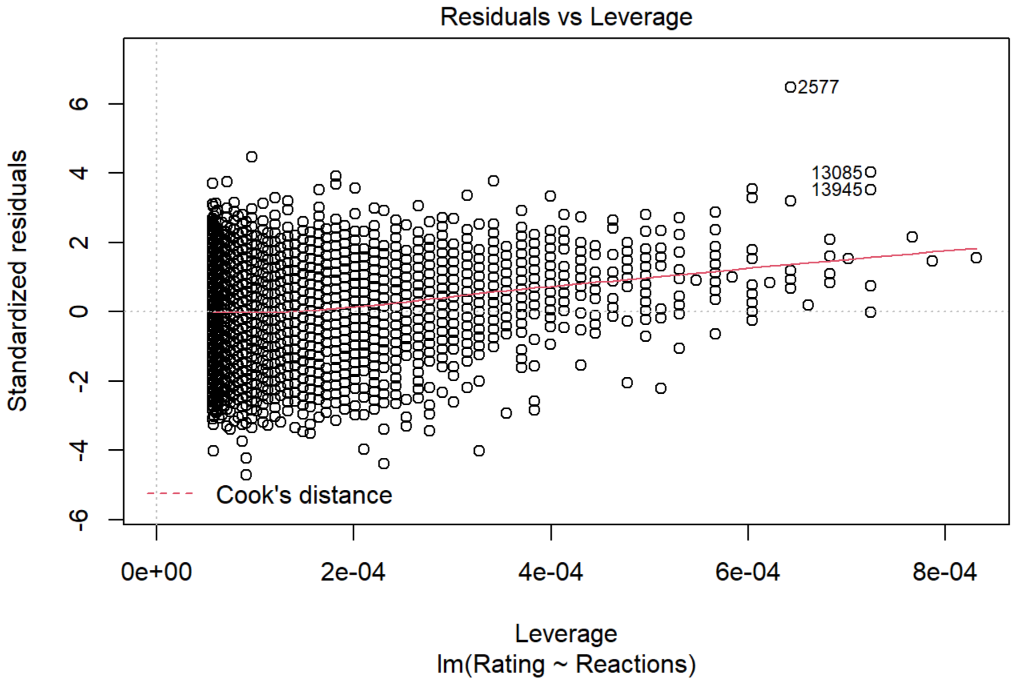



```
plot(lmFitSimple)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Rating ~ Reactions)
2577
623
17229
-4
-2
0
2
4
6


Scale-Location
√|Standardized residuals|
Fitted values
lm(Rating ~ Reactions)
2577
623
17229
0.0
0.5
1.0
1.5
2.0
2.5
50
60
70
80



Porównując z regresja nieliniową dla optymalnego wielomianu 9 stopnia:

```
lmFit9 <- lm(Rating ~ poly(Reactions, 9))
anova(lmFitSimple, lmFit9)
```

```
## Analysis of Variance Table
##
## Model 1: Rating ~ Reactions
## Model 2: Rating ~ poly(Reactions, 9)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1  17586 276878
## 2  17578 269558  8    7319.4 59.663 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Wybrane ciekawsze analizy poszczególnych parametrów

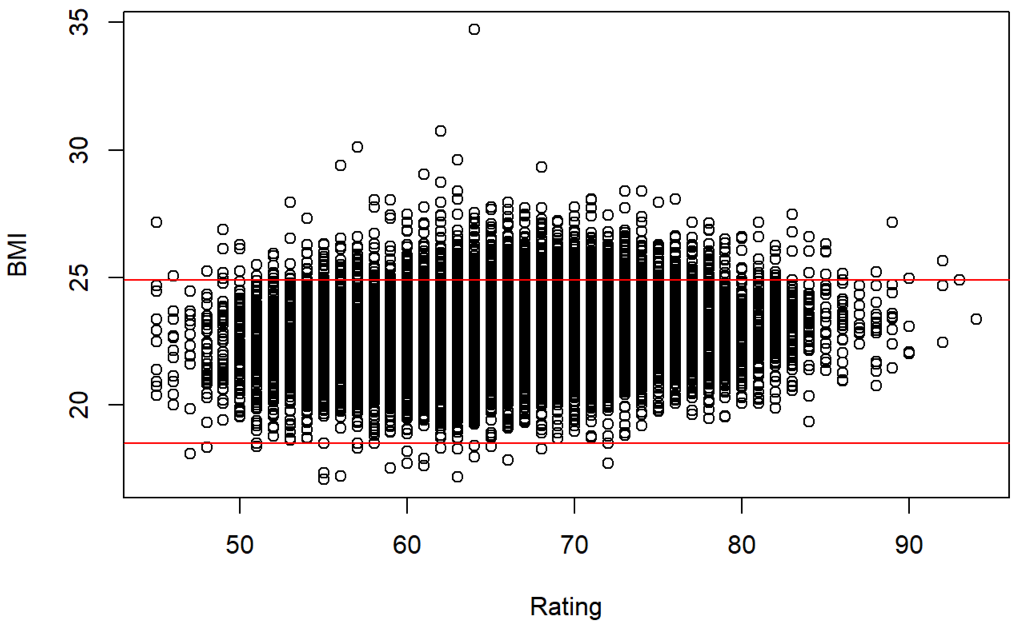
BMI piłkarzy

Przy pomocy parametrów wzrostu i wagi wyznaczono BMI dla każdego zawodnika.

Na wykresie przedstawiono wartość BMI dla rankigu zawodników. Na czerwono zaznaczono granice wartości określanych jako “in the healthy weight range”.

```
BMI = Weight / (Height / 100)^2

plot(Rating, BMI)
abline(h=18.5, col="red")
abline(h=24.9, col="red")
```



Wnioski:

- Bardziej prawdopobonbe jest, że profesjonalny piłkarz ma naddwagę niż niedowagę
- Najlepsi piłkarze w zdecydowanej większości znajdują się w zakresie zdrowej wartości BMI.

Pozycja zawodnika a parametry jakościowe

W następnej kolejności zbadano jak pozycja na jakiej występuje zawodnik ma wpływ na jego wybrane parametry. Celem było zbadanie, czy w rzeczywistości na pewnych pozycjach niezbędne jest posiadanie odpowiednich cech.

Szybkość

Piewszym znanym twierdzeniem jest, teza, że skrzydłowy musi być przede wszystkim szybki. Postanowiono zbadać czy zebrane dane potwierdzają taką tezę.

W tym celu kolumnę zbadano średnią i medianę wartości *Speed* w zależności od kolumny *Club_position*.

```
position_speed <- data.frame(Club_Position, Speed)
mean_speed = aggregate(.~Club_Position, data=position_speed, mean)
print(head(mean_speed[order(-mean_speed[,2]),], 10))
```

```
##    Club_Position    Speed
## 16            LW 80.09774
## 27            RW 79.35338
## 25            RM 77.10145
## 14            LM 76.55797
## 28           RWB 74.35556
## 17           LWB 74.31111
## 8            LAM 74.30556
## 19            RB 74.21898
## 9             LB 73.76503
## 18           RAM 73.38889
```

```
median_speed = aggregate(.~Club_Position, data=position_speed, median)
print(head(median_speed[order(-median_speed[,2]),], 10))
```

```
##    Club_Position Speed
## 27            RW  80.0
## 16            LW  79.0
## 25            RM  78.0
## 14            LM  77.0
## 8            LAM  75.0
## 17           LWB  75.0
## 19            RB  75.0
## 9             LB  74.0
## 28           RWB  74.0
## 18           RAM  73.5
```

Wyjaśnienia powyższych skrótóW:

- LW - Left Winger
- RW - Right Winger
- RM - Right Midfielder
- LM - Left Midfielder
- RWB - Right Winger Back
- LWB - Left Winger Back
- LAM - Left Attacking Midfielder
- RB - Right Back
- LB - Left Back
- RAM - Right Attacking Midfielder

Powyższe zestawienie potwierdza, że boczni zawodnicy, w szczególności skrzydłowi powinni cechować się znaczną szybkością.

Wzrost

Oczywistym jest, że najwyższym wzrostem powinni cechować się bramkarze. Inne znane tezy mówią, że wysocy powinni być również środkowi obrońcy. Czasami mówi się również, że wzrost może być dużym atutem naspastników. W celu sprawdzenia takiej zależności przeprowadzono analizę analogiczną jak poprzednio jednak z wykorzystaniem parametru *Height* zamiast *Speed*.

```
position_height <- data.frame(Club_Position, Height)
mean_height = aggregate(.~Club_Position, data=position_height, mean)
print(head(mean_height[order(-mean_height[,2]),], 5))
```

```
##    Club_Position   Height
## 1                189.0000
## 7             GK 188.4130
## 3             CB 186.4337
## 20           RCB 186.1912
## 10           LCB 186.0143
```

```
median_height = aggregate(.~Club_Position, data=position_height, median)
print(head(median_height[order(-median_height[,2]),], 5))
```

```
##    Club_Position Height
## 1                   189
## 7             GK    188
## 3             CB    187
## 10           LCB    186
## 20           RCB    186
```

Powyższe wyniki potwierdzają, że w czołówce pozycji pod względem wzrostu znajdują się bramkarze, oraz środkowi obrońcy (CB - Center Back)

Wytrzymałość

Często mówi się, że najwięcej biegać muszą defensywni pomocy. W celu weryfikacji czy w parametrach zawodników można znaleźć taką zależność przeprowadzono analizę poziomu wytrzymałości w zależności od pozycji zajmowanej na boisku.

```
position_stamina <- data.frame(Club_Position, Stamina)
mean_stamina = aggregate(.~Club_Position, data=position_stamina, mean)
print(head(mean_stamina[order(-mean_stamina[,2]),], 5))
```

```
##    Club_Position  Stamina
## 17           LWB 75.91111
## 22           RDM 75.71805
## 19            RB 75.32664
## 12           LDM 75.28195
## 21           RCM 75.00283
```

```
median_stamina = aggregate(.~Club_Position, data=position_stamina, median)
print(head(median_stamina[order(-median_stamina[,2]),], 5))
```

```
##    Club_Position Stamina
## 4            CDM      77
## 22           RDM      77
## 12           LDM      76
## 19            RB      76
## 21           RCM      76
```

Również te wyniki potwerdziły słuszność wcześniej wymienionej tezy.

Największą wytrzymałością cechowali się zawodnicy na pozycji DM (Defensive Midfielder).